Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083047

ARTICLE

Decentralized Sports Streaming Authorization: A Three-Layer Cryptographic Architecture for Live and On-Demand Access

Liangyu Lin, Li Feng* and Lin Huang

Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, China

*Corresponding Author: Li Feng. Email: lfeng@must.edu.mo

Received: 02 April 2026; Accepted: 08 May 2026; Published: 15 June 2026

ABSTRACT: The modern sports streaming market is severely fragmented, forcing fans into costly, siloed platforms. While blockchain-based decentralized architectures offer a unified, interoperable sport streaming ecosystem, securely delivering commercial video over untrusted infrastructure remains a profound cryptographic challenge. Existing schemes fail to simultaneously support highly granular on-demand highlights and large scale dynamic live subscriptions. To resolve this, we propose a novel decentralized authorization architecture that systematically integrates existing cryptographic primitives into a decoupled three-layer protocol. By securely bridging on-chain state transitions with off-chain cryptographic enforcement, our architecture directly maps commercial payment workflows onto the underlying key evolution logic. The protocol leverages a top-down Binary Hash Tree for arbitrary-grained aggregated authorization of static highlights, while combining a salt-governed forward hash chain with dynamic Identity-Based Broadcast Encryption to achieve state-driven adaptive authorization for live streams. Rigorous theoretical analysis and extensive evaluations confirm the robust security and high efficiency of our proposed architecture.

KEYWORDS: Sports streaming media; decentralized authorization; end-to-end encrypted access control; revocation; blockchain

1 Introduction

The global sports streaming market [1] is experiencing unprecedented fragmentation. To maximize revenue, top-tier sports leagues are splitting their media rights across various independent platforms, forcing fans into a costly and siloed viewing experience where following a single team often requires multiple subscriptions. Concurrently, the sports consumption paradigm has fundamentally evolved. Modern audiences demand not only traditional Live Subscriptions [2] (e.g., full-match broadcasts or season passes) but also highly granular on-demand Highlights [3–5] (e.g., short customized clips, player-specific moments, and secondary creations).

To unify this fragmented landscape, blockchain technologies [6] offer a highly promising solution. By leveraging blockchain as a trustless settlement layer and employing wallet-backed self-sovereign identities as universal identities [7], a decentralized ecosystem can dismantle traditional platform silos. This paradigm allows fans to access content from multiple independent broadcasters, or even different federations (e.g., the National Basketball Association (NBA), Premier League, and the Federation Internationale de Football Association (FIFA)), through a single, interoperable portal. It restores a seamless viewing experience for users while enabling content providers to directly monetize diverse digital sporting assets within a transparent, decentralized media economy.

Realizing this unified ecosystem requires delivering commercial video over untrusted infrastructure, such as decentralized storage networks or public content delivery networks [6,8]. To protect copyrights, the video payload must be rigorously end-to-end encrypted [9]. However, it introduces a profound challenge: the selective sharing of encrypted data. This challenge is complex in sports streaming, where data consumption operates along two distinct dimensions: historical Highlights and Live Subscriptions. Highlights demand retroactive, fine-grained isolation of bounded historical intervals (e.g., a specific 10-s goal clip) from a massive, pre-encrypted stream. Conversely, Live Subscriptions demand open-ended, continuous access to future data as it is generated, coupled with the ability to instantly sever access for departing viewers.

Most cryptographic data-sharing schemes are ill-equipped to handle these heterogeneous access patterns. Traditional approaches, such as relying on public-key encryption or Attribute-Based Encryption (ABE) [10], inherently force access policies to be hard-coded into ciphertexts. This static paradigm collapses in sports streaming, where the live audience is highly volatile, and eventual highlight buyers are entirely unknown during the initial broadcast. Furthermore, even advanced underlying cryptographic structures fall short. Binary Hash Trees (BHT) excel at static historical boundaries but cannot scale for continuous live subscriptions. Conversely, hash-chain-based subscription models natively support continuous access but suffer from an "all-or-none" limitation, failing to enforce exact historical boundaries for short clips. To this end, Droplet [11] proposed a unified architecture combining both primitives, but its continuous key derivation exposes a critical "subscribe-cancel-resubscribe" vulnerability that leaks unauthorized data. While recent schemes like SegSub [12] mitigate this by segmenting the regression chain, this rigid truncation fundamentally destroys the ultra-fine-grained expressiveness required for precise highlight clipping. More critically, when confronting the massive audience churn of live sports, both architectures collapse: their reliance on point-to-point (unicast) key updates inevitably triggers crippling $O(N)$ communication storms. Ultimately, no existing framework can simultaneously fulfill the stringent security and efficient authorization requirements of both live subscriptions and on-demand highlights.

To bridge this gap, instead of designing a fundamentally new cryptographic primitive, we propose a decentralized authorization architecture built upon a novel decoupled three-layer cryptographic protocol. The core novelty lies in the system design: rather than forcing access policies into static ciphertexts, our architecture ingeniously orchestrates existing primitives to physically isolate heterogeneous access demands into distinct layers. For static content, it utilizes a top-down BHT exclusively for payload encryption, fulfilling the ultra-fine-grained isolation demanded by on-demand highlights without introducing data redundancy. Conversely, for highly volatile live subscriptions, it orchestrates a salt-governed forward hash chain alongside dynamic Identity-Based Broadcast Encryption (IBBE). Under this mechanism, on-chain revocation events trigger the injection of unpredictable cryptographic salts to explicitly sever the forward derivation path for departing viewers. Simultaneously, dynamic IBBE securely encapsulates these salt updates into a single $O(1)$ broadcast, allowing all remaining legitimate subscribers to seamlessly maintain their access.

In summary, our main contributions are:

- A decentralized authorization architecture is proposed to securely bridge on-chain state transitions with off-chain cryptographic enforcement. By directly mapping commercial payment workflows onto the underlying key evolution logic, it establishes a transparent media economy enabling seamless, cross-platform, and cross-federation interoperability, allowing users to access diverse sports streams through a single portal.
- A novel three-layer protocol design unifies heterogeneous access demands. This architecture design elegantly integrates existing cryptographic primitives into a single underlying cryptographic foundation. For on-demand highlights, it achieves arbitrary-grained aggregated authorization, enabling exact historical clipping without data redundancy. Concurrently, for continuous live streaming, it realizes

state-driven adaptive authorization to execute dynamic, real-time access control with robust security and high efficiency.

- Rigorous theoretical proofs demonstrate that the proposed scheme inherently guarantees forward and backward secrecy, alongside robust resistance to complex "subscribe-cancel-resubscribe" vulnerabilities. Furthermore, extensive evaluations confirm the architecture's overall efficiency across both consumption paradigms, demonstrating optimal communication overhead for exact-chunk highlight retrieval and strict $O(1)$ server scalability during live audience churn, with client latency remaining safely within standard buffer limits.

For clarity and ease of reference, the principal notations used throughout our system design and analysis are summarized in Table 1.

Table 1: Main notation.

Symbol	Meaning
m_i	Video chunk with global index i.
k_i	Content Encryption Key (CEK) for chunk m_i (acts as a static data encryption key).
s_i	Temporal state corresponding to chunk m_i (acts as live Subscriber Encryption Key (SEK)).
C_i	Static data ciphertext of chunk m_i.
C_i^*	Dynamic key-ciphertext wrapping k_i for live access.
ρ_i	Block height of the most recent on-chain revocation event.
η_{ρ_i}	Fresh Verifiable Random Function (VRF) salt at block height ρ_i.
A_{ρ_i}	Accumulator digest representing the active subscriber set.
W_{u,ρ_i}	Cryptographic witness of legitimate user u at block height ρ_i.
$\mathcal{N}_{a,b}$	Minimum node cover derived from the BHT for chunk range $[a, b]$.
d	Depth of the Layer 1 BHT.
$\|I\|$	Length of the requested historical highlight interval.
K	Total number of video chunks in a continuous live stream.
N	Total number of active subscribers in the ecosystem.
R	Number of revoked users in a single revocation batch.

2 Related Work

Recent research relevant to our problem spans three main areas: blockchain-enabled media ecosystems, sports highlight generation, and cryptographic data sharing.

2.1 Blockchain-Enabled Media Distribution and Decentralized Access Control

Blockchain has emerged as a coordination layer for transparent settlement and direct media monetization, though most designs focus on coarse-grained business workflows rather than chunk-level stream authorization [1]. While prototypes successfully integrate smart contracts with decentralized storage for live video delivery [6], practical InterPlanetary File System deployments often face non-trivial performance and centralization trade-offs [8,13,14]. Similarly, decentralized access control frameworks for cloud and Internet of Things systems [15–17], alongside self-sovereign identity mechanisms [7,18], effectively reduce central authority reliance. However, they typically manage access at the file or service level, failing to address the dual demands of fine-grained historical clipping and continuous live subscriptions within a single encrypted stream.

2.2 Sports Highlights and Personalized Consumption

Multimedia research has significantly advanced automatic sports highlight generation [3], personalized detection [4], implicit user-specified generation [5], and automated narration [19]. While these works underscore the growing demand for highly granular, personalized sports clips, they primarily focus on content understanding and generation. They generally assume the desired footage is already accessible to the system, leaving the orthogonal challenge of cryptographically enforcing fine-grained paid access over untrusted distribution networks unresolved.

2.3 Cryptographic Data Sharing for Historical and Live Access

ABE offers expressive fine-grained authorization [10,20], but hard-codes policies into ciphertexts, rendering it inflexible for live streams with dynamically changing audiences. For temporal access, interval-based schemes efficiently delegate bounded historical ranges [21], while key regression enables forward-only derivation of future decryption states [22]. Droplet combined these concepts into a unified framework for decentralized streams [11], and SegSub further improved resilience against re-subscription leakage via chain segmentation [12]. Additionally, accumulators and broadcast encryption have been utilized for scalable revocation and group updates [23,24]. Despite these advances, no prior system simultaneously delivers exact-granularity historical clipping, secure forward-only live continuation, resistance to subscribe-cancel-resubscribe abuse, and server-side $O(1)$ communication during massive audience churn—a gap our three-layer architecture precisely bridges. Table 2 summarizes the key differences between representative cryptographic data-sharing schemes and our proposed design.

Table 2: Comparison of cryptographic data-sharing schemes.

Scheme	Fine-Grained Access	Continuous Stream Authorization	Revocation Overhead	SCR Security
Traditional ABE [10,20]	Limited; (hard-coded into ciphertexts)	Not applicable	$O(N)$ re-encryption	Not applicable
Droplet [11]	Yes	Yes	$O(N)$ rekeying storm under mass revocation	Vulnerable
SegSub [12]	No (Segment-level access)	Yes	Grouping collapses; surges to $O(N)$	Secure
Our scheme	**Yes; Arbitrary exact-chunk access**	**Yes; Adaptive authorization**	***O* (1)**	**Secure**

Note: N denotes the number of active subscribers; SCR stands for Subscribe-Cancel-Resubscribe.

3 Preliminaries

In this section, we define the three foundational cryptographic primitives utilized in our architecture.

3.1 Binary Hash Trees

BHT [25] is a hierarchical structure designed for deterministic key derivation. It employs a top-down approach to expand a single root secret into a large set of related keys. Let $S_{0,1} \in \{0,1\}^{\lambda}$ be the root seed at depth 0, where λ represents the security parameter. Using two independent one-way hash functions H_L and

H_R, the nodes at depth $d+1$ are recursively derived from their parent node $S_{d,i}$ at depth d as follows:

$$S_{d+1,2i-1} = H_L(S_{d,i}), \quad S_{d+1,2i} = H_R(S_{d,i}) \tag{1}$$

where $i \in \{1, \ldots, 2^d\}$ denotes the node index at depth d. For a tree of depth k, this construction yields $N = 2^k$ distinct leaf keys. The BHT structure inherently guarantees logarithmic delegation and cryptographic isolation, providing a highly efficient mechanism for hierarchical authorization in decentralized environments.

3.2 Dynamic IBBE

Dynamic IBBE [24] is a cryptographic primitive that securely transmits a message M to a dynamically changing set of authorized receivers $\mathcal{U}_t$ at any given time t. Built upon the standard Rivest-Shamir-Adleman cryptographic framework, the system computes a succinct group digest A_t to represent the current authorized set, and issues a private decryption key (or witness) sk_u^t to each user $u \in \mathcal{U}_t$. To distribute content, a broadcast ciphertext CT_t is generated as follows:

$$CT_t = \mathsf{IBBE.Encrypt}(\mathsf{PK}, A_t, M), \tag{2}$$

where PK denotes the global public parameters. Any valid user $u \in \mathcal{U}_t$ can process CT_t using their specific key sk_u^t and the digest A_t to recover M. When the authorized set changes, the system simply updates A_t and the corresponding public parameters to reflect the new membership.

4 Decentralized Streaming Authorization System Design

In this section, we present the architecture of our decentralized authorization system, designed to support the full lifecycle of sports streaming. At its core, the proposed scheme decouples the *transaction plane* (trading and state management), the *control plane* (dynamic access management), and the *data plane* (static payload encryption). By integrating the blockchain with a novel three-layer cryptographic protocol, our design mathematically maps the commercial payment logic directly onto the underlying key evolution logic. Consequently, the system achieves fine-grained, dynamic authorization with robust security and low overhead. We first formalize the system model, and subsequently detail the proposed three-layer cryptographic protocol.

4.1 System Model

We consider a global ecosystem comprising multiple independent sports leagues and federations (e.g., the Premier League, the NBA, and the FIFA) acting as content providers. Each provider broadcasts numerous live matches to a global audience. Aligning with standard modern streaming protocols such as Hypertext Transfer Protocol Live Streaming and Moving Picture Experts Group Dynamic Adaptive Streaming over Hypertext Transfer Protocol, a continuous live match video $\mathcal{V}$ is digitized and segmented into a finite sequence of chronological data chunks, denoted as $\mathcal{V} = \{m_1, m_2, \ldots, m_N\}$. To protect commercial value, these chunks are subsequently encrypted, cached, and distributed via untrusted infrastructure, including both decentralized storage networks and content delivery networks.

To securely orchestrate this ecosystem, our system involves the following interacting entities, as illustrated in Fig. 1:

- **Content Provider (CP):** The CP is the owner and producer of the commercial streaming data. It orchestrates the entire authorization lifecycle by registering matches and commercial parameters (e.g., pricing models) on the blockchain, continuously encrypting the live video chunks, and enforcing flexible access control across diverse consumption models. Given the industrial scale of global sports broadcasting, we

assume the CP is equipped with robust computational infrastructure (e.g., dedicated servers or cloud environments) capable of handling high-definition video processing and real-time cryptography.

- **Live Subscribers:** Live subscribers are the end-users dedicated to accessing the real-time broadcast (e.g., via a season pass or a pay-per-view live match). They initiate on-chain transactions to purchase a continuous *state of access over time* from the CP. Given the massive scale of global sports audiences, their authorization state is inherently dynamic (e.g., frequently joining or revoking mid-stream), directly driving the rapid and continuous evolution of the active authorized set $\mathcal{U}_t$.
- **Highlight Buyers:** Highlight buyers are the end-users who purchase specific video segments post-match (e.g., a 10-s goal clip or a player highlight). They initiate on-chain transactions to acquire on-demand access to a discrete chunk range. This long-tail, fragmented consumption pattern necessitates strictly granular authorization.
- **Blockchain:** The blockchain is the decentralized ledger for financial settlement. It maintains the global authorization state via a smart contract (SC) layer, functioning as programmable, autonomously executing state machines. Both content providers (e.g., diverse sports leagues) and global end-users register on-chain, leveraging cryptographic wallet addresses as universal identities to achieve seamless cross-platform interoperability.
- **Storage Infrastructure:** Serving purely as the data delivery pipeline, this infrastructure is strictly modeled under the honest-but-curious security assumption. While it faithfully caches and routes the encrypted data, it remains completely oblivious to the plaintext content, cryptographic keys, and authorization logic. This explicitly decouples the ecosystem's security from infrastructural trust.

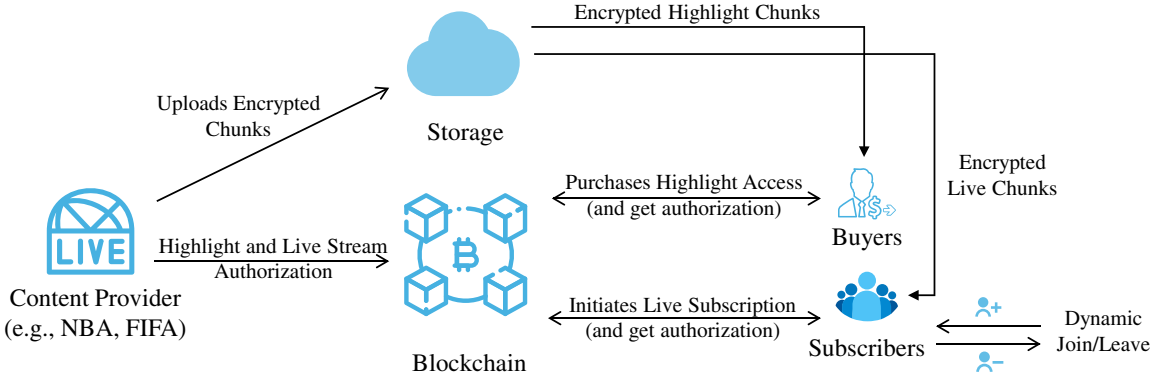

Figure 1: System model of the decentralized sports streaming authorization architecture. The ecosystem decouples the transaction plane (Blockchain), the data plane (Storage), and the control plane to serve both Live Subscribers and Highlight Buyers.

4.2 Three-Layer Cryptographic Protocol

To achieve extremely fine-grained authorization (down to a single chunk m_i) while executing highly scalable dynamic access control, we propose a three-layer cryptographic architecture. This design decouples the static data payload from the dynamic authorization state by partitioning the cryptographic operations into chunk-level static encryption, stream-level key derivation, and group-level access control, thereby eliminating data re-encryption and redundant ciphertexts. The decoupled architecture of the first two layers is illustrated in Fig. 2.

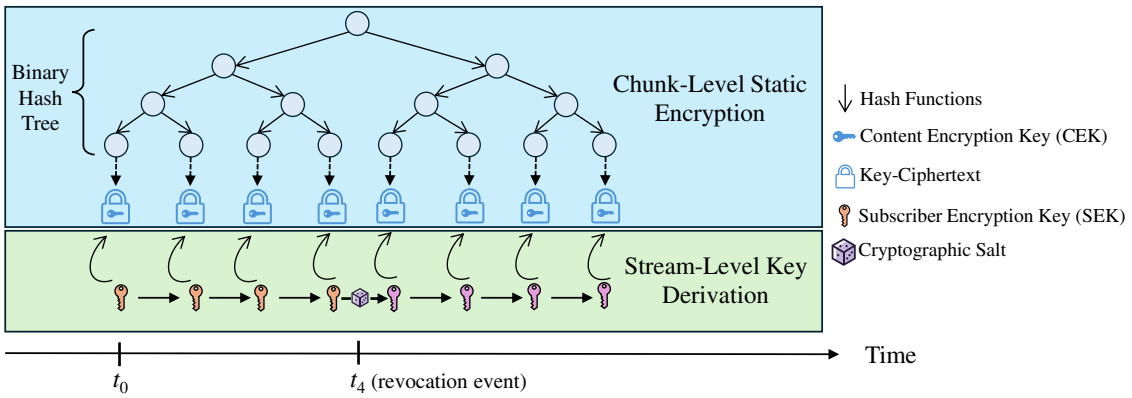

Figure 2: The decoupled cryptographic architecture of Layer 1 and Layer 2. Layer 1 (top) utilizes a BHT for static CEK derivation. Layer 2 (bottom) maintains a temporal hash chain of SEKs. During a revocation event (e.g., at t_4), a fresh cryptographic salt is injected to break the forward derivation chain, naturally enforcing access severance without re-encrypting the payload.

Layer 1: Chunk-Level Static Encryption

For a raw video stream comprising a chronological sequence of discrete chunks $\mathcal{V} = \{m_1, m_2, \ldots, m_K\}$, the layer executes independent, chunk-by-chunk encryption. By configuring a BHT with an adequate depth $d \geq \log_2 K$, we deterministically derives a mutually independent CEK k_i for each respective chunk m_i. Consequently, the raw payload is strictly symmetrically encrypted exactly once, yielding the data ciphertext:

$$C_i = \mathsf{SymEnc}(k_i, m_i), \tag{3}$$

which is subsequently offloaded to the storage infrastructure. Benefiting from the BHT's top-down key derivation, this layer natively enables *arbitrary-grained key aggregation*. This topological property allows the system to efficiently share any arbitrary highlight clip of the sports stream (e.g., aggregating a minimal set of intermediate tree nodes to authorize a discrete chunk range), explicitly fulfilling the highlight buyers' demands without introducing data redundancy.

Layer 2: Stream-Level Key Derivation

For the chronological delivery of a live streaming, this layer executes a continuous, time-bound key encapsulation. By initializing a unidirectional hash chain, we recursively derives a temporal state s_i uniquely corresponding to each chunk m_i. This derivation is governed by a dynamic cryptographic salt, denoted as η_i, defined mathematically as $s_i = H(s_{i-1} \parallel \eta_i)$. Consequently, rather than exposing the static CEK k_i directly, the CP symmetrically encrypts it using the current temporal state s_i, yielding the lightweight key-ciphertext:

$$C_i^* = \mathsf{SymEnc}(s_i, k_i) \tag{4}$$

which is continuously distributed as the composite ciphertext tuple $\{C_i, C_i^*\}$. To further detail the state transition, let ρ_i denote the block height of the most recent on-chain revocation event synchronized by the CP prior to computing state s_i. The salt η_i is updated by

$$\eta_i = \begin{cases} 0, & \text{if } \rho_i = \rho_{i-1} \\ \mathsf{VRF}(sk_{CP}, \rho_i), & \text{if } \rho_i > \rho_{i-1} \end{cases} \tag{5}$$

where VRF denotes the VRF evaluation parameterized by the CP's secret key sk_{CP}.

During periods of stable on-chain membership ($\rho_i = \rho_{i-1}$), the salt defaults to a deterministic zero ($\eta_i = 0$). Legitimate subscribers holding a valid prior state s_{i-1} can autonomously derive s_i and subsequent states strictly via local hash operations (i.e., $s_i = H(s_{i-1} \parallel 0)$).

Conversely, upon detecting a new on-chain revocation ($\rho_i > \rho_{i-1}$), the CP transitions the authorization phase. By injecting a verifiable, unpredictable salt η_i, the CP mathematically breaks the hash derivation chain. Lacking this parameter, the revoked user is precluded from deriving s_i and any future states. Meanwhile, the remaining valid subscribers securely receive the updated η_i (detailed in Layer 3) to resume their autonomous derivation. Due to the one-way nature of the hash chain, this layer naturally guarantees backward secrecy, which means that newly joined subscribers cannot rewind the hash progression to decrypt past video chunks.

In this manner, Layer 2 achieves an event-triggered adaptive authorization segmentation, which enforces precise, time-bound access control without ever requiring the heavy static payloads in Layer 1 to be re-encrypted.

Layer 3: Group-Level Access Control

To securely distribute the dynamic salt generated in Layer 2 exclusively to authorized subscribers, this layer establishes a group-level access control mechanism by leveraging a Dynamic IBBE scheme.

Let $\mathcal{U}_{\rho_i}$ denotes the set of active, authorized blockchain addresses, and let A_{ρ_i} represent its corresponding accumulator value bound to the latest revocation block height ρ_i. The layer encapsulates the newly derived salt (denoted as η_{ρ_i}) into a broadcast ciphertext:

$$\hat{C}_{\rho_i} = \mathsf{IBBE.Enc}(\mathsf{PP}, A_{\rho_i}, \eta_{\rho_i}) \tag{6}$$

where PP denotes the global public parameters. This ciphertext $\hat{C}_{\rho_i}$ is subsequently packaged into an on-chain transaction. The legitimate subscriber $u \in \mathcal{U}_{\rho_i}$ utilizes a cryptographic witness W_{u,ρ_i} to retrieve the salt as follows:

$$\eta_{\rho_i} = \mathsf{IBBE.Dec}(W_{u,\rho_i}, \hat{C}_{\rho_i}). \tag{7}$$

Cryptographically, the witness W_{u,ρ_i} is inherently bound to the accumulator state A_{ρ_i}. When the authorized group changes, the accumulator A_{ρ_i} is updated, requiring remaining legitimate subscribers to update their witness locally via the standard Stateless Witness Update algorithm [26]:

$$W_{u,\rho_i} = \mathsf{WitnessUpdate}(W_{u,\rho_{i-1}}, \Delta\boldsymbol{\mathcal{R}}_{\rho_i}, \mathsf{PP}) \tag{8}$$

where $\Delta\mathcal{R}_{\rho_i}$ denotes the membership delta (e.g., the set of revoked and added addresses) in the authorized group, which can be derived directly from recorded on-chain transactions. Users excluded from A_{ρ_i} cannot obtain a valid witness to decrypt $\hat{C}_{\rho_i}$, inherently guaranteeing *forward secrecy*. Additionally, regardless of the sheer volume of addresses in $\mathcal{U}_{\rho_i}$, the size of $\hat{C}_{\rho_i}$ depends solely on the public parameters PP, thus ensuring an $O(1)$ communication overhead.

Ultimately, this three-layer architecture establishes a rigorous cryptographic foundation tailored for decentralized sports streaming. Specifically, Layer 1 achieves arbitrary-grained aggregated authorization, empowering users to unlock customized, discrete video segments (e.g., highlights) using a single compact key token. Concurrently, Layer 2 and Layer 3 jointly realize state-driven adaptive authorization for continuous live subscriptions.

4.3 Arbitrary-Grained Highlight Authorization

In the on-demand scenario, the live event has concluded, and the entire video $\mathcal{V}$ is securely anchored within the static Layer 1 BHT. A viewer v wishes to purchase a specific highlight reel spanning from chunk m_a to m_b. Since the temporal stream has finalized, this paradigm bypasses the Layer 2 Hash Chain and Layer 3 IBBE entirely, relying strictly on a direct on-chain delivery mechanism. The authorization protocol proceeds in four stages (as summarized in Algorithm 1):

Step 1: Purchase Initiation. Viewer v triggers a smart contract function BuyHighlight $(addr_v, a, b)$, transferring the requisite cryptocurrency. Upon payment validation, the contract emits a public PurchaseRequest event.

Step 2: Minimum Cover Derivation. The CP monitors the blockchain for purchase events. To authorize access, the CP derives the *Minimum Node Cover* $\mathcal{N}_{a,b}$ from the BHT, which represents the minimal set of higher-level intermediate nodes required to perfectly reconstruct the requested subtree.

Step 3: Token Delivery. To ensure secure delivery over the public ledger, the CP encrypts $\mathcal{N}_{a,b}$ using the viewer's public key pk_v. The resulting cryptographic token is $Token_v = \mathsf{AsymEnc}(pk_v, \mathcal{N}_{a,b})$. The CP submits $Token_v$ to the smart contract via a fulfillment transaction, prompting the contract to emit the token compactly within an on-chain event log.

Step 4: Decryption. Viewer v retrieves the emitted token from the blockchain event log, decrypts $Token_v$ using their local wallet private key sk_v, and deterministically derives all underlying CEKs $\{k_j\}_{j=a}^{b}$ via top-down hashing to unlock the specific video chunks.

Algorithm 1: Arbitrary-grained highlight authorization

Require: Smart Contract SC, Content Provider CP, Viewer v (keys pk_v, sk_v), chunk range $[a, b]$
Ensure: Keys $\{k_j\}_{j=a}^{b}$ for specific highlight chunks

// 1. On-Chain Purchase Phase
▷ Viewer v initiates payment
SC.BuyHighlight$(addr_v, a, b)$
if Payment Validated **then**
    SC.Emit(PurchaseRequest, $addr_v, a, b$)
end if
// 2. Token Derivation & Delivery Phase
▷ CP derives BHT cover
$\mathcal{N}_{a,b} \leftarrow$ CP.MinNodeCover(a, b)
$Token_v \leftarrow \mathsf{AsymEnc}(pk_v, \mathcal{N}_{a,b})$
▷ CP fulfills order via SC
SC.Emit(TokenEvent, $Token_v$)
// 3. Client Decryption Phase
$Token_v \leftarrow$ SC.FetchEventLog()
▷ Viewer v unwraps token
$\mathcal{N}_{a,b} \leftarrow \mathsf{AsymDec}(sk_v, Token_v)$
▷ Top-down key derivation
$\{k_j\}_{j=a}^{b} \leftarrow$ BHT.Derive$(\mathcal{N}_{a,b})$
return $\{k_j\}_{j=a}^{b}$

4.4 State-Driven Live Subscription

Unlike the static on-demand scenario, live sports streaming requires continuous temporal progression and dynamic membership management. This workflow activates the full three-layer architecture to handle the onboarding of new subscribers and the access severance of revoked users, entirely without re-encrypting the video payload. The protocol operates through a continuous steady-state derivation, punctuated by rigorous state transitions, detailed in Algorithm 2:

Step 1: Subscription Initiation. When a new viewer v subscribes mid-match, the smart contract dynamically adds $addr_v$ to the authorized group and updates the accumulator to A' to reflect this addition. As defined in Layer 2, the current temporal salt η_{ρ_i} remains unchanged, since salt rotation is strictly triggered by revocations (detailed below).

Step 2: Token Delivery. When the CP detects the on-chain update, it extracts the current Layer 2 temporal state s_j alongside generating the initial Layer 3 witness W_v. The CP encrypts this minimal state tuple using the viewer's public key as $Token_v = \mathsf{AsymEnc}(pk_v, \{W_v, s_j\})$, and submits it to the contract to emit compactly via an event log.

Step 3: Derivation and Decryption. Viewer v retrieves and decrypts $Token_v$ to obtain $\{W_v, s_j\}$. With s_j acquired, the viewer immediately begins playback. As the CP continuously broadcasts the encrypted chunks and key-ciphertexts $C^*_{j+1} = \mathsf{SymEnc}(s_{j+1}, k_{j+1})$ generated in layer 2, the viewer autonomously rolls their local hash chain $s_{j+1} = H(s_j \parallel 0)$ to unwrap the underlying video keys. The witness W_v is securely stored locally, remaining dormant until the next revocation triggers a salt rotation.

Revocation: When a user w is revoked (e.g., subscription expiration) at the subsequent block height ρ_{i+1}, the CP triggers a strict state transition to sever access.

(a) The CP updates the accumulator to $A_{\rho_{i+1}}$, permanently excluding user w.
(b) It generates a fresh salt $\eta_{\rho_{i+1}} = \mathsf{VRF}(sk_{CP}, \rho_{i+1})$, mathematically breaking the forward derivation path from the previous state.
(c) It broadcasts the new ciphertext $\hat{C}_{\rho_{i+1}} = \mathsf{IBBE.Enc}(\mathsf{PP}, A_{\rho_{i+1}}, \eta_{\rho_{i+1}})$.

Excluded from $A_{\rho_{i+1}}$, the revoked user w cannot obtain a valid witness to decrypt $\hat{C}_{\rho_{i+1}}$. Consequently, their local hash chain stalls, explicitly preventing them from deriving any subsequent temporal states. Meanwhile, remaining legitimate viewers autonomously update their witness to $W_{v,\rho_{i+1}}$ locally using public blockchain records, decrypt the new salt $\eta_{\rho_{i+1}}$, and continuously derive subsequent video keys.

Algorithm 2: State-driven live subscription and revocation

Require: Smart Contract SC, Content Provider CP, Viewer v, Public Params PP
Ensure: Continuous state derivation and dynamic revocation
// 1. Subscription & Steady-State Phase
▷ Updates accumulator to A'
1: $\mathsf{SC.AddSubscriber}(addr_v)$
2: $W_v \leftarrow \mathsf{CP.GenWitness}(A')$
▷ CP emits token via SC
3: $Token_v \leftarrow \mathsf{AsymEnc}(pk_v, \{W_v, s_j\})$
▷ Viewer v unwraps token
4: $\{W_v, s_j\} \leftarrow \mathsf{AsymDec}(sk_v, Token_v)$
5: **while** no revocation occurs **do**
▷ Viewer v autonomously rolls hash chain

(Continued)

Algorithm 2 (continued)

$s_{j+1} \leftarrow H(s_j \parallel 0)$
▷ Unwraps underlying video keys
$k_{j+1} \leftarrow \mathsf{SymDec}(s_{j+1}, C^*_{j+1})$
end while
// 2. Revocation & State Transition Phase (at block ρ_{i+1})
▷ Updates accumulator to $A_{\rho_{i+1}}$
$\mathsf{SC.Revoke}(addr_w)$
▷ CP injects fresh salt
$\eta_{\rho_{i+1}} \leftarrow \mathsf{VRF}(sk_{CP}, \rho_{i+1})$
▷ CP broadcasts new salt
$\hat{C}_{\rho_{i+1}} \leftarrow \mathsf{IBBE.Enc}(\mathsf{PP}, A_{\rho_{i+1}}, \eta_{\rho_{i+1}})$
// 3. Legitimate Viewer Recovery
for all remaining legitimate viewer v **do**
$W_{v,\rho_{i+1}} \leftarrow \mathsf{WitnessUpdate}(W_{v,\rho_i}, \Delta\mathcal{R}_{\rho_{i+1}}, \mathsf{PP})$
$\eta_{\rho_{i+1}} \leftarrow \mathsf{IBBE.Dec}(W_{v,\rho_{i+1}}, \hat{C}_{\rho_{i+1}})$
▷ Resumes forward derivation
$s_{k+1} \leftarrow H(s_k \parallel \eta_{\rho_{i+1}})$
end for

5 Security and Performance Analysis

In this section, we theoretically analyze the proposed architecture, demonstrating its rigorous cryptographic security guarantees, alongside an evaluation of its performance.

5.1 Adversarial Model

We model the adversary $\mathcal{A}$ as a Probabilistic Polynomial-Time (PPT) entity. The adversary can observe all on-chain transactions and event logs, control the honest-but-curious storage infrastructure, obtain all public ciphertexts $\{C_i, C_i^*\}$, and adaptively issue subscription, cancellation, resubscription, and highlight-purchase queries through corrupted blockchain addresses. The adversary may also pool all credentials legitimately obtained by corrupted identities, including historical BHT tokens, stale temporal states, and stale IBBE witnesses. However, $\mathcal{A}$ cannot compromise the CP's signing key, VRF secret key, or IBBE master secret, cannot violate the finality and correct execution of the underlying blockchain, and cannot force an honest active subscriber to redistribute freshly decrypted salts, keys, or plaintext outside the protocol.

5.2 Formal Security Definitions

Definition 1 (Forward Secrecy): *We define forward secrecy via a challenge-response game G_{FS} between a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$. Let t_c be the time of a cancellation query made by $\mathcal{A}$. $\mathcal{A}$ wins if it can distinguish the temporal state s_t (where $t > t_c$) from a truly random string with a non-negligible advantage. The proposed architecture guarantees forward secrecy if, for all PPT adversaries, the advantage in G_{FS} is negligible.*

Definition 2 (Backward Secrecy): *Symmetrically, we define backward secrecy via a game G_{BS}. Let t_s be the time of a subscription query. $\mathcal{A}$ wins if it can distinguish any historical temporal state s_t (where $t < t_s$) from a random string with a non-negligible advantage.*

Definition 3 (SCR Security under Collusion): *We formalize Subscribe-Cancel-Resubscribe (SCR) security under a collusion model. Let $\mathcal{A}$ adaptively issue a cancellation query at t_1 and a subsequent resubscription query*

at t_2 ($t_1 < t_2$). By pooling the stale credentials from t_1 and the fresh credentials from t_2, $\mathcal{A}$ attempts to derive the symmetric key k_t for any chunk within the unauthorized gap $t \in [t_1, t_2 - 1]$. The architecture achieves SCR security if $\mathcal{A}$'s advantage in deriving s_t remains negligible despite the pooled knowledge.

5.3 Security Proofs

Theorem 1 (Forward Secrecy): *The proposed architecture ensures forward secrecy such that a revoked adversary $\mathcal{A}$ at chunk t_c cannot derive any subsequent temporal state s_t for $t > t_c$.*

Proof of Theorem 1: Suppose $\mathcal{A}$ wins G_{FS} by deriving s_{t_c+1}. According to the protocol, $s_{t_c+1} = H(s_{t_c} \parallel \eta_\rho)$. While $\mathcal{A}$ may possess the stale state s_{t_c}, computing s_{t_c+1} strictly requires the fresh salt η_ρ, which is encapsulated within the broadcast ciphertext $\hat{C}_\rho$ using the dynamic IBBE scheme. Since $\mathcal{A}$'s witness W_u is mathematically excluded from the updated IBBE accumulator A_ρ upon revocation, deriving η_ρ is equivalent to breaking the underlying IBBE scheme's indistinguishability under chosen-plaintext attack. Furthermore, the pseudo-randomness of the VRF ensures η_ρ cannot be predicted without the secret key sk_{CP}. Thus, $\mathcal{A}$'s advantage is negligible. □

Theorem 2 (Backward Secrecy): *The proposed architecture ensures backward secrecy such that a newly joined adversary $\mathcal{A}$ at chunk t_s cannot derive any historical temporal state s_t for $t < t_s$.*

Proof of Theorem 2: Let $\mathcal{A}$ possess the current temporal state s_{t_s} upon subscription. To win G_{BS}, $\mathcal{A}$ must derive s_{t_s-1} from the known $s_{t_s} = H(s_{t_s-1} \parallel \eta_{t_s})$. This requires inverting the cryptographic hash function H. Under the assumption that H exhibits pre-image resistance, finding s_{t_s-1} is computationally infeasible for any PPT adversary. By induction, all historical states $\{s_0, \ldots, s_{t_s-1}\}$ remain hidden. Thus, $\mathcal{A}$'s advantage is negligible. □

Theorem 3 (SCR Attack Resistance and Collusion Resilience): *The architecture is resilient against SCR attacks and collusion among multiple corrupted identities, ensuring that pooled credentials yield no advantage in unauthorized intervals.*

Proof of Theorem 3: Consider an extreme collusion scenario (satisfying Definition 3) where $\mathcal{A}$ pools credentials from a revoked session at t_1 and a new session at t_2 ($t_1 < t_2$). Let the pooled knowledge be $\mathcal{K}_{collusion} = \mathcal{K}_{t_1} \cup \mathcal{K}_{t_2}$. To derive any key k_t for the unauthorized gap $t \in [t_1, t_2 - 1]$, $\mathcal{A}$ must either roll s_{t_1} forward or s_{t_2} backward. As demonstrated in Theorem 1, the forward derivation is blocked by the exclusion from the IBBE-protected salt update. As demonstrated in Theorem 2, the backward derivation is blocked by the pre-image resistance of the hash chain. Since both the forward and backward cryptographic paths are severed at the authorization boundaries, combining the knowledge $\mathcal{K}_{t_1}$ and $\mathcal{K}_{t_2}$ provides no algebraic path to s_t. Thus, the architecture is secure against both SCR and collusion attacks. □

Remark 1 (Out-of-Band Key Sharing and Piracy): *While our architecture rigorously prevents cryptographic collusion (i.e., attackers attempting to derive unauthorized keys by pooling stale and fresh credentials, as formally proven in Theorem 3), it does not prevent a legitimate, active subscriber from voluntarily redistributing their correctly decrypted keys or plaintext video payload to unauthorized users. Such out-of-band key sharing is fundamentally beyond the scope of cryptographic access control models. In commercial digital rights management deployments, mitigating this specific piracy vector requires integrating complementary orthogonal technologies, such as digital watermarking or traitor tracing schemes, which can be seamlessly applied to the Layer 1 static payloads without altering our proposed three-layer authorization protocol.*

5.4 Complexity Analysis

By decoupling the static video payload from dynamic access rights, our architecture achieves highly scalable performance strictly bounded by constants or logarithmic functions.

Communication Overhead. For continuous live subscriptions, the Layer 3 dynamic IBBE ensures that the broadcast ciphertext $\hat{C}_{\rho_{i+1}}$ size remains strictly constant at $O(1)$, entirely independent of the total subscriber population $|\mathcal{U}|$. For arbitrary-grained highlight authorization, the encrypted token uniquely encapsulates the minimum node cover $\mathcal{N}_{a,b}$. Due to the structural properties of the Layer 1 BHT, the token size is rigorously bounded by $O(\log|I|)$, where M is the requested chunk span. These properties make on-chain event emissions exceptionally economical and scalable.

Computational Efficiency (Content Provider). During steady-state streaming, the CP's workload is minimal, requiring only one symmetric encryption and one hash evaluation $O(1)$ per video chunk. For the most intensive administrative operation (e.g., dynamic revocation), the CP only computes the accumulator state transition and generates a single $O(1)$ IBBE ciphertext $\hat{C}_{\rho_{i+1}}$.

Computational Efficiency (Subscriber). Authorized viewers execute real-time playback on local devices with negligible overhead. Unwrapping a temporal chunk requires precisely one $O(1)$ hash derivation and one symmetric decryption. Crucially, during membership transitions, remaining legitimate viewers locally execute the witness update. The complexity of the stateless witness update algorithm is mathematically bounded by $O(\min(|\Delta\mathcal{R}|, |\mathcal{U}_{active}|))$, where $\mathcal{U}_{active}$ denotes the set of remaining legitimate viewers. For standard sparse revocations, the overhead scales incrementally with $O(|\Delta\mathcal{R}|)$. However, during massive revocation events where the revoked population exceeds a specific threshold, the algorithm efficiently reconstructs the witness based on the shrinking active population, causing the computational overhead to decrease. This guarantees that end-users can seamlessly maintain temporal hash progression under any extreme membership volatility, even on resource-constrained mobile devices.

6 Performance Evaluation

In this section, we evaluate the performance of our scheme through three core dimensions: communication efficiency for historical highlights, server-side scalability under high churn, and the client-side latency trade-off. We compare Our Scheme against two decentralized stream sharing schemes, Droplet [11] and SegSub [12], under workloads that reflect the high-churn and fine-grained access patterns typical of decentralized sports streaming.

6.1 Efficiency of On-Demand Highlights

To evaluate the communication overhead for historical video highlights, we simulate a stream of $K = 4096$ chunks using Secure Hash Algorithm 256-bit (SHA-256) (32-Byte keys). We configure SegSub with a segment length of $l = 16$ and account for Droplet's 33-Byte privacy-preserving ephemeral key overhead. Fig. 3 evaluates the communication cost across varying highlight lengths from 10 to 1000 chunks under two conditions: *Relaxed Granularity* and *Exact-Chunk Granularity*.

Boundary Alignment and Privacy Overhead. As shown in Fig. 3, both Our Scheme and Droplet exhibit jagged curves. This is because the communication cost in BHT-based designs depends on how well the requested intervals align with the tree structure. Although both schemes achieve an $O(\log|I|)$ bound, Droplet has a consistently higher cost due to its 33-Byte "privacy tax" for identity-unlinkability. Our Scheme removes this extra payload because such extreme anonymization is not required for sports streaming. As a result, Our Scheme achieves the most efficient overhead for exact-chunk authorization.

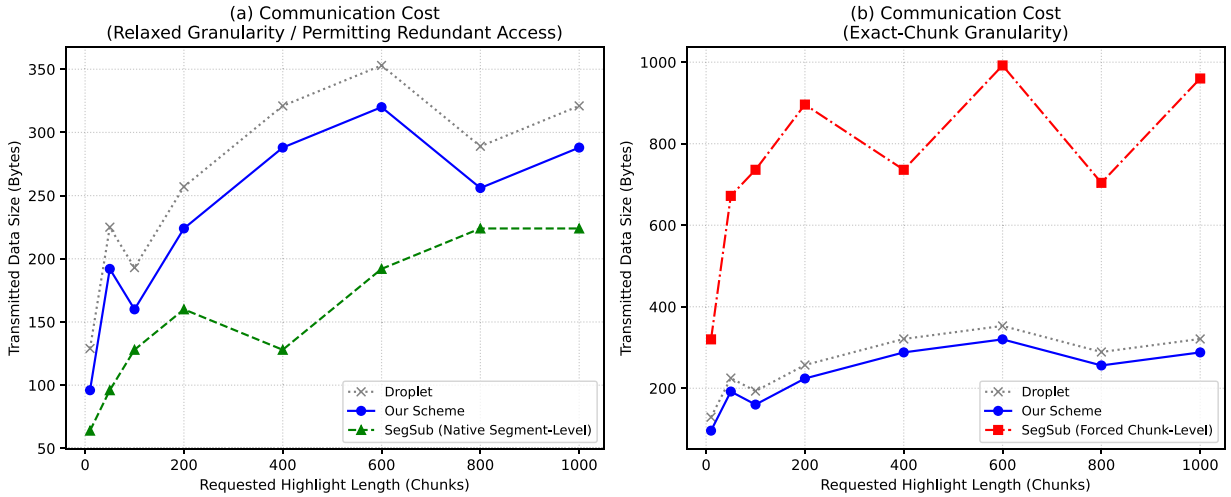

Figure 3: Communication overhead for on-demand highlight authorization. (**a**) Evaluated under relaxed granularity, permitting segment-level redundant access. (**b**) Evaluated under strict exact-chunk granularity, ensuring zero data leakage.

Impact of Access Precision. As shown in Fig. 3a, SegSub seems competitive under relaxed granularity because it delivers keys at the segment level. However, this approach is limited because it often includes extra, unrequested chunks within those segments to simplify key delivery. This means the provider either risks data leakage by sending more than requested or faces higher costs to be precise. When we enforce *Exact-Chunk Granularity* to prevent such leakage (Fig. 3b), SegSub's performance drops significantly. This happens because SegSub send many individual 32-Byte keys for any chunks that fall outside its segment boundaries. In contrast, Our Scheme supports precise access for any interval from the start, which keeps the communication cost low while strictly following the principle of least privilege.

6.2 Server Communication Overhead under Revocation

To evaluate scalability under high churn, we examine the server communication overhead during batch revocation events. Fig. 4 plots the server communication cost (in kilobytes) relative to the number of revoked users (R), assuming a total audience of $N = 100{,}000$.

Droplet relies on a unicast mechanism to distribute updated keys to all remaining valid users, causing its communication cost to scale linearly with $N - R$. As shown in Fig. 4, its overhead starts extremely high and decreases only when the majority of the user base is revoked. SegSub attempts to optimize this by grouping users (e.g., $M = 100$ per group). For very small R, it successfully reduces communication by broadcasting to unaffected groups. However, as R increases, the probability of a group containing at least one revoked user rapidly approaches 1. Once a group is "tainted" by a revoked user, its shared group key becomes invalid, forcing SegSub to fall back to unicasting new keys to the remaining valid users in that specific group. Consequently, when R grows (e.g., $R \geq 1000$), SegSub's grouping mechanism collapses, and its overhead surges to match Droplet's linear cost.

In contrast, Our Scheme eliminates the need for user-specific key distribution. By leveraging IBBE and on-chain state tracking, the server only needs to broadcast a single, constant-size epoch ciphertext. As illustrated by the flat blue line in Fig. 4, the server communication overhead remains strictly $O(1)$ and minimal, providing absolute immunity to revocation storms regardless of the batch size R. However, this $O(1)$ communication efficiency involves an inherent architectural trade-off by shifting the cryptographic

burden to the clients. Consequently, Our Scheme incurs a higher client-side computation latency than Droplet and SegSub (Fig. 5), although this localized state-transition overhead is safely absorbed within the 2000 ms decoding deadline of standard chunk buffers (Fig. 6).

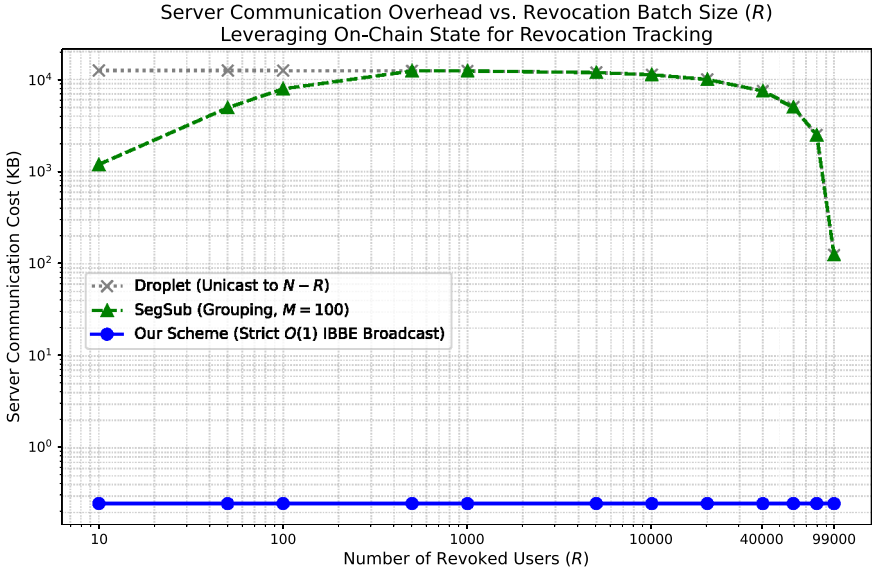

Figure 4: Server communication overhead relative to the revocation batch size R. SegSub degrades to Droplet's unicast cost as R increases, while Our Scheme maintains the $O(1)$ broadcast cost.

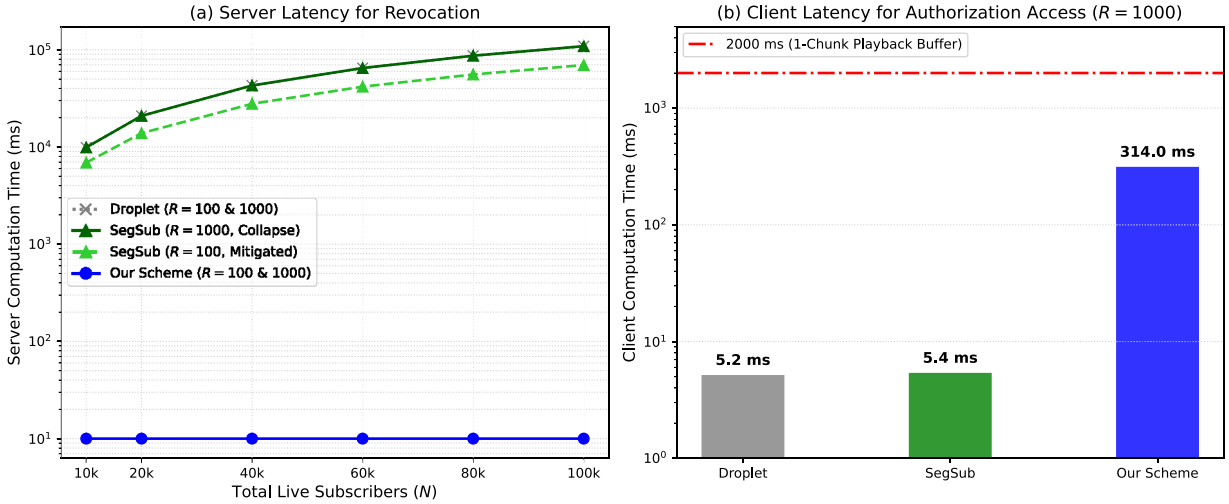

Figure 5: End-to-end computation latency trade-off during a major revocation event ($R = 1000$, $N = 100{,}000$). (**a**) Server-side latency (log scale) demonstrating our scheme's absolute immunity. (**b**) Client-side authorization latency, illustrating that our scheme's decentralized witness computation safely consumes less than 16% of the standard 2000 ms chunk buffer.

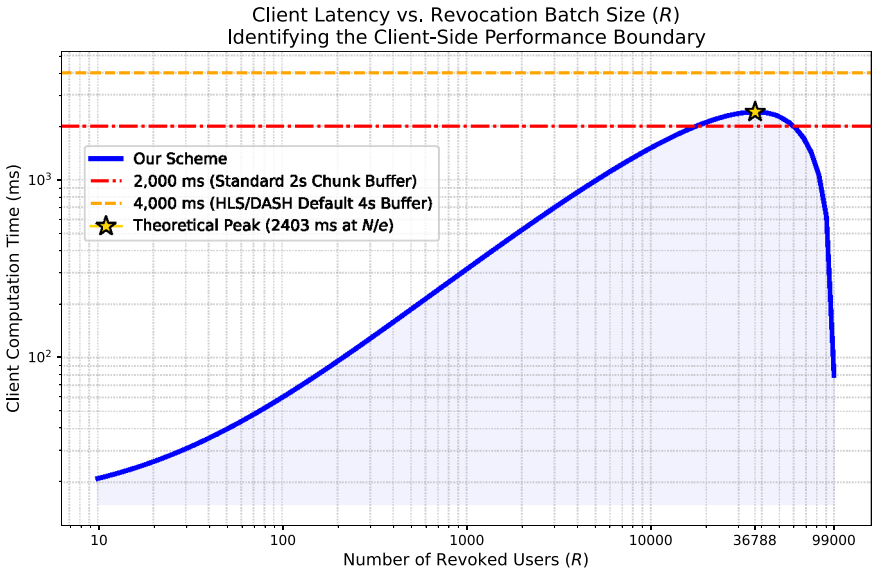

Figure 6: Client computation latency relative to the revocation batch size R, identifying the theoretical performance boundary at $R = N/e$.

6.3 End-to-End Computation Latency Trade-off

To evaluate practical deployment, we anchor our simulation parameters to standard micro-benchmarks. For the server, we model a standard cloud instance (Elliptic Curve Integrated Encryption Scheme: 1.1 ms, SHA-256: 1.2 µs, IBBE broadcast: 10 ms). For the client, we model a typical smartphone (Elliptic Curve Digital Signature Algorithm: 4.4 ms, SHA-256: 45 µs, IBBE resolve: 15 ms).

Server-Side Scalability. During a batch revocation of $R = 1000$ out of N = 100,000 users, Droplet and SegSub require the server to perform nearly 99,000 asymmetric encryptions. This sequential load exceeds 1.5 min, forcing the live stream to stall. In contrast, Our Scheme offloads witness updates to the clients, requiring the server to execute only a single $O(1)$ IBBE encryption. As shown in Fig. 5a, the server latency remains strictly at 10.0 ms, rendering it immune to revocation storms.

Client-Side Latency. To achieve this server scalability, legitimate clients should fetch the on-chain revocation list and locally compute the $O(R \log_2(N/R))$ witness update. Fig. 5b shows that this computation takes approximately 314 ms for $R = 1000$. While higher than the baselines' 5 ms, this 314 ms latency safely consumes less than 16% of the standard 2000 ms decoding buffer margin typical of chunk-based streaming protocols.

Theoretical Boundary and Practical Limitations. Since the client latency is dominated by local hashing, we evaluate its non-linear scaling to identify hardware limits. The computational complexity reaches its theoretical maximum at $R = N/e$ (≈36,787 evictions for $N = 100{,}000$), resulting in a latency of 2403 ms. While this marginally exceeds the 2000 ms buffer limit (Fig. 6), it is practically mitigated by two factors. First, the probability of an instantaneous 36% churn rate within a 2-s epoch is negligible; real-world churn is distributed over time, keeping R safely below this boundary. Second, this peak represents a transient, one-time state synchronization cost. Once updated, the latency for subsequent chunks instantly reverts to the nominal baseline, resulting in at most a sub-second stutter isolated to a single chunk. By trading a transient local delay for the absolute eradication of $O(N)$ server-side bottlenecks, the system achieves genuine scalability.

Since the IBBE decryption overhead is constant (15 ms), the client latency is entirely dominated by the local batch witness update, which scales mathematically at $O(R\log_2(N/R))$. Fig. 6 tracks this non-linear scaling behavior.

Remark 2 (Mitigating Blockchain Latency): *Our system is immune to blockchain latency during steady-state operations, as subscribers autonomously roll the hash chain off-chain ($\eta = 0$). Latency is only introduced during revocation events, which require waiting for block confirmations to retrieve a fresh salt η. For seamless playback, this latency can be mitigated via: (1) Scheduled Revocation: The new salt's activation is scheduled for a future chunk, absorbing confirmation delays with negligible economic impact (e.g., granting revoked users brief extra access); and (2) Pre-Confirmation Propagation: The CP immediately distributes the updated ciphertext $\hat{C}$ via the peer-to-peer network, allowing subscribers to verify the signature and retrieve η without waiting for strict on-chain finality.*

Remark 3 (Client-Side Revocation Overhead): *The primary client-side computational bottleneck stems from the IBBE witness updates triggered by membership revocations. As illustrated in Figs. 5 and 6, this overhead is manageable under typical churn rates. For constrained clients (e.g., low-power devices), this computation can be practically absorbed without playback stuttering by expanding the playback buffer (e.g., from 2 s to 4 s) or dynamically adjusting the video frame rate. Furthermore, to address increased natural churn at a massive system scale, our future research will explore reputation-based tiered grouping. By isolating volatile users, this approach minimizes the revocation scale within stable clusters, fundamentally reducing the witness update burden for the majority of clients.*

7 Conclusion

This paper presents a decentralized architecture that unifies the fragmented sports streaming ecosystem. By decoupling static video payloads from dynamic access rights via a three-layer cryptographic protocol, we directly map commercial workflows onto the underlying key evolution. Our approach successfully resolves the technical conflict between fine-grained highlight isolation and continuous live streaming authorization. We proved the system's strict forward and backward secrecy over untrusted infrastructure. Furthermore, by permanently eradicating $O(N)$ communication bottlenecks and maintaining $O(1)$ server scalability under massive audience churn, this framework establishes a secure, highly efficient, and interoperable foundation for the future Web3 sport streaming economy.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Liangyu Lin and Li Feng; methodology, Liangyu Lin; software, Lin Huang; validation, Li Feng; formal analysis, Li Feng; investigation, Lin Huang; writing—original draft preparation, Liangyu Lin; writing—review and editing, Liangyu Lin; visualization, Lin Huang; project administration, Li Feng. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Berkani AS, Moumen H, Benharzallah S, Yahiaoui S, Bounceur A. Blockchain use cases in the sports industry: a systematic review. Int J Netw Distrib Comput. 2024;12(1):17–40. doi:10.1007/s44227-024-00022-3.

2. Zhou K, Wu T, Zhang J. EGOP: a server-side enhanced architecture to eliminate end-to-end latency caused by GOP length in live streaming. Comput Mater Contin. 2026;86(1):1–27.
3. Midoglu C, Sabet SS, Sarkhoosh MH, Majidi M, Gautam S, Solberg HM, et al. AI-based sports highlight generation for social media. In: Proceedings of the 3rd ACM Mile-High Video Conference; 2024 Feb 11–14; Denver, CO, USA. p. 7–13.
4. Chen R, Zhou P, Wang W, Chen N, Peng P, Sun X, et al. PR-Net: preference reasoning for personalized video highlight detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 7980–9.
5. Kim M, Lee D, Noh J. Generating highlight videos of a user-specified length using most replayed data. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems; 2025 Apr 26–May 1; Yokohama, Japan. p. 1138:1–13.
6. Lopes EJ, Kataria S, Keshav S, Ikram ST, Ghalib MR, Shankar A, et al. Live video streaming service with pay-as-you-use model on ethereum blockchain and interplanetary file system. Wirel Netw. 2022;28(7):3111–25. doi:10.1007/s11276-022-03009-6.
7. Ding Y, Yu J, Li S, Sato H, Machizawa MG. Data aggregation management with self-sovereign identity in decentralized networks. IEEE Trans Netw Serv Manag. 2024;21(6):6174–89. doi:10.1109/tnsm.2024.3451995.
8. Wu Z, Yang CR, Vargas S, Balasubramanian A. Is IPFS ready for decentralized video streaming? In: Proceedings of the ACM Web Conference 2023; 2023 Apr 30–May 4; Austin, TX, USA. p. 3002–10.
9. Alluhaidan ASD, Prabu P. End-to-end encryption in resource-constrained IoT device. IEEE Access. 2023;11:70040–51. doi:10.1109/access.2023.3292829.
10. Zhang Y, Deng RH, Xu S, Sun J, Li Q, Zheng D. Attribute-based encryption for cloud computing access control: a survey. ACM Comput Surv. 2021;53(4):83:1–41. doi:10.1145/3398036.
11. Shafagh H, Burkhalter L, Ratnasamy S, Hithnawi A. Droplet: decentralized authorization and access control for encrypted data streams. In: Proceedings of the 29th USENIX Security Symposium (USENIX Security 20); 2020 Aug 12–14; Virtual. p. 2469–86.
12. Gao S, Zhao Q, Li G, Feng L, Shen M, Zhang P, et al. SegSub: balancing security and efficiency for large-scale decentralized data subscription in Web3. IEEE Trans Netw Sci Eng. 2026;13:4261–76.
13. Trautwein D, Raman A, Tyson G, Castro I, Scott W, Schubotz M, et al. Design and evaluation of IPFS: a storage layer for the decentralized web. In: Proceedings of the ACM SIGCOMM 2022 Conference; 2022 Aug 22–26; Amsterdam, The Netherlands. p. 739–52.
14. Balduf L, Korczyński M, Ascigil O, Keizer NV, Pavlou G, Scheuermann B, et al. The cloud strikes back: investigating the decentralization of IPFS. In: Proceedings of the 2023 ACM Internet Measurement Conference; 2023 Oct 24–26; Montreal, QC, Canada. p. 391–405.
15. Pal S, Dorri A, Jurdak R. Blockchain for IoT access control: recent trends and future research directions. J Netw Comput Appl. 2022;203:103371.
16. Wei X, Yan Y, Guo S, Qiu X, Qi F. Secure data sharing: blockchain-enabled data access control framework for IoT. IEEE Internet Things J. 2022;9(11):8143–53.
17. Qin B, Liu J, Xing X, Meng W, Liu Y. Mitigating centralization in access control system with blockchain and distributed storage. In: Proceedings of the 2024 IEEE International Conference on Blockchain (Blockchain); 2024 Aug 19–22; Copenhagen, Denmark. p. 340–5.
18. Vrielynck PJ, hamme TV, Ghostin R, Lagaisse B, Preuveneers D, Joosen W. A self-sovereign identity approach to decentralized access control with transitive delegations. In: Proceedings of the 29th ACM Symposium on Access Control Models and Technologies; 2024 May 15–17; San Antonio, TX, USA. p. 139–47.
19. Sarfati N, Yerushalmy I, Chertok M, Keller Y. Generating factually consistent sport highlights narrations. In: Proceedings of the 6th International Workshop on Multimedia Content Analysis in Sports; 2023 Oct 29; Ottawa, ON, Canada. p. 15–22.
20. Li X, Yang G, Xiang T, Xu S, Zhao B, Deng RH, et al. Make revocation cheaper: hardware-based revocable attribute-based encryption. In: Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP); 2024 May 19–23; San Francisco, CA, USA. p. 3109–27.

21. Crampton J. Practical and efficient cryptographic enforcement of interval-based access control policies. ACM Trans Inf Syst Secur. 2011;14(1):14:1–30. doi:10.1145/1952982.1952996.
22. Fu K, Kamara S, Kohno T. Key regression: enabling efficient key distribution for secure distributed storage. Cryptology ePrint archive. Paper 2005/303. 2005 [cited 2026 May 7]. Available from: https://eprint.iacr.org/2005/303.
23. Loporchio M, Bernasconi A, Maesa DDF, Ricci L. A survey of set accumulators for blockchain systems. Comput Sci Rev. 2023;49(2014):100570. doi:10.1016/j.cosrev.2023.100570.
24. Delerablée C. Identity-based broadcast encryption with constant size ciphertexts and private keys. In: Advances in Cryptology–ASIACRYPT 2007. Vol. 4833 of Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer; 2007. p. 200–15.
25. Kumar V, Petit J, Whyte W. Binary hash tree based certificate access management for connected vehicles. In: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks; 2017 Jul 18–20; Boston, MA, USA. p. 145–55.
26. Camenisch J, Lysyanskaya A. Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Advances in Cryptology–CRYPTO 2002. Vol. 2442 of Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer; 2002. p. 61–76.